



## Разработка сервиса обмена данными между программными системами

- К. Д. Тандалов** студент, лаборант-исследователь НИЛ-211 «Искусственный интеллект в производственных системах»; Самарский национальный исследовательский университет имени академика С. П. Королёва, г. Самара; tkd186@mail.ru
- В. А. Печенин** кандидат технических наук, доцент, заведующий НИЛ-211 «Искусственный интеллект в производственных системах»; Самарский национальный исследовательский университет имени академика С. П. Королёва, г. Самара; pechenin.va@ssau.ru

*В данной статье представлена разработка информационной модели сервиса для обмена данными между ERP- и MES-системами, используемыми для планирования и контроля производственных процессов. Целью сервиса является эффективная передача и структурирование информации между базами данных в производственном процессе, а также контроль операций через пользовательский интерфейс. Такой подход позволяет обеспечить интеграцию различных внешних программных систем, что является актуальной задачей в области автоматизации и управления производственными процессами. Реализация сервиса осуществлена на Python с использованием библиотеки работы с базами данных SQLAlchemy. Графический интерфейс сервиса позволяет отредактировать необходимые JSON-шаблоны, используемые для работы с базами данных. Было проведено тестирование на двух базах данных для системы мониторинга движения деталей в процессе производства, подтверждающее функциональность и надёжность сервиса.*

**Ключевые слова:** информационная модель; система; JSON; база данных; Python

**Цитирование:** Тандалов, К. Д. Разработка сервиса обмена данными между программными системами / К. Д. Тандалов, В. А. Печенин // Динамика и виброакустика. – 2024. – Т. 10, №3. – С. 50-57. DOI: 10.18287/2409-4579-2024-10-3-50-57

---

### Введение

В настоящее время формируется база VI технологического уклада в экономиках развитых стран и ряда развивающихся стран. Формирование технологической базы VI уклада является важным вызовом для Российской экономики. На данном этапе промышленная политика направлена на технологическую модернизацию производства [1].

Для планирования ресурсов на предприятиях используются ERP-системы, для контроля за исполнением плана – MES-системы, в которых собирается и структурируется необходимая информация. Для аэрокосмической отрасли характерны увеличенные циклы изготовления продукции. Имеется необходимость принятия своевременных корректирующих мероприятий при планировании, а для этого требуется повышение объективности при контроле движения материальных потоков. MES-системы, к сожалению, не отменяют ошибки, связанные с человеческим фактором.

Именно преодоление разрыва в данных при планировании и актуальном состоянии производства является важной проблемой на сегодня. Собранные данные передаются из MES в ERP [2]. Передача данных в MES может происходить и из сторонних программ [3], имеющих отдельные базы данных (БД), для снижения ошибок, связанных с человеческим фактором. В работе описана разработанная информационная модель сервиса загрузки-выгрузки данных между программными системами, выполняющая функции конвертера данных между программами планирования и сбора данных.

## 1 Разработка информационной модели сервиса загрузки-выгрузки данных

Идея представленного сервиса между программными системами заключается в обеспечении эффективного обмена информацией между ERP- и MES-системами. Целью такого сервиса является сбор, структурирование и передача информации между различными БД, а также контролирование этих операций через пользовательский интерфейс, чтобы обеспечить процесс интеграции систем контроля в производственный процесс предприятия. Таким образом, данный сервис является связующим звеном для систем сбора и анализа данных. На рисунке 1 приведена схема разработанной информационной модели передачи данных.



Рисунок 1 – Схема сервиса для обработки и передачи данных

В сервисе существует три модуля: модуль интерфейса пользователя, модуль проведения операций и модуль работы с базами данных. Модуль для работы с базами данных загружает содержимое двух сравниваемых баз, используя дополнительно информацию по набору таблиц и их полей, загружаемую из файла формата \*.json. В модуле проведения операций производятся последовательное сравнение таблиц и операции добавления, удаления и обновления информации. Обновлённая информация передается в модуль работы с БД, и производится запись обратно в базу данных системы анализа.

## 2 Реализация сервиса

Сервис был реализован на языке программирования Python, поскольку тот обеспечивает удобство, гибкость и имеет широкое применение в разработке подобных сервисов для обмена и обработки информации.

Для реализации модуля работы с БД была использована библиотека SQLAlchemy. SQLAlchemy предоставляет высокоуровневый интерфейс для работы с данными, позволяя разработчикам работать через объектно-реляционное отображение (ORM) или через низкоуровневые SQL-выражения.

Благодаря модульности и расширяемости решения количество сущностей и полей в базах данных не является ограничивающим фактором, так как использованы универсальные классы для создания ORM моделей. Например, класс InitModel динамически формирует структуру таблиц на основе переданных данных, что позволяет адаптировать базу данных под различные типы данных и их комбинации. Каждая соответствующая своей таблице ORM модель (из моделей model\_0, model\_1 ... model\_15) наследует от соответствующего базового класса (Base2), определяя свою уникальную структуру данных и методы для взаимодействия с ними. Этот подход обеспечивает гибкость при разработке и позволяет легко добавлять новые сущности и поля без необходимости изменения общей архитектуры приложения.

На рисунке 2 приведён интерфейс программы.

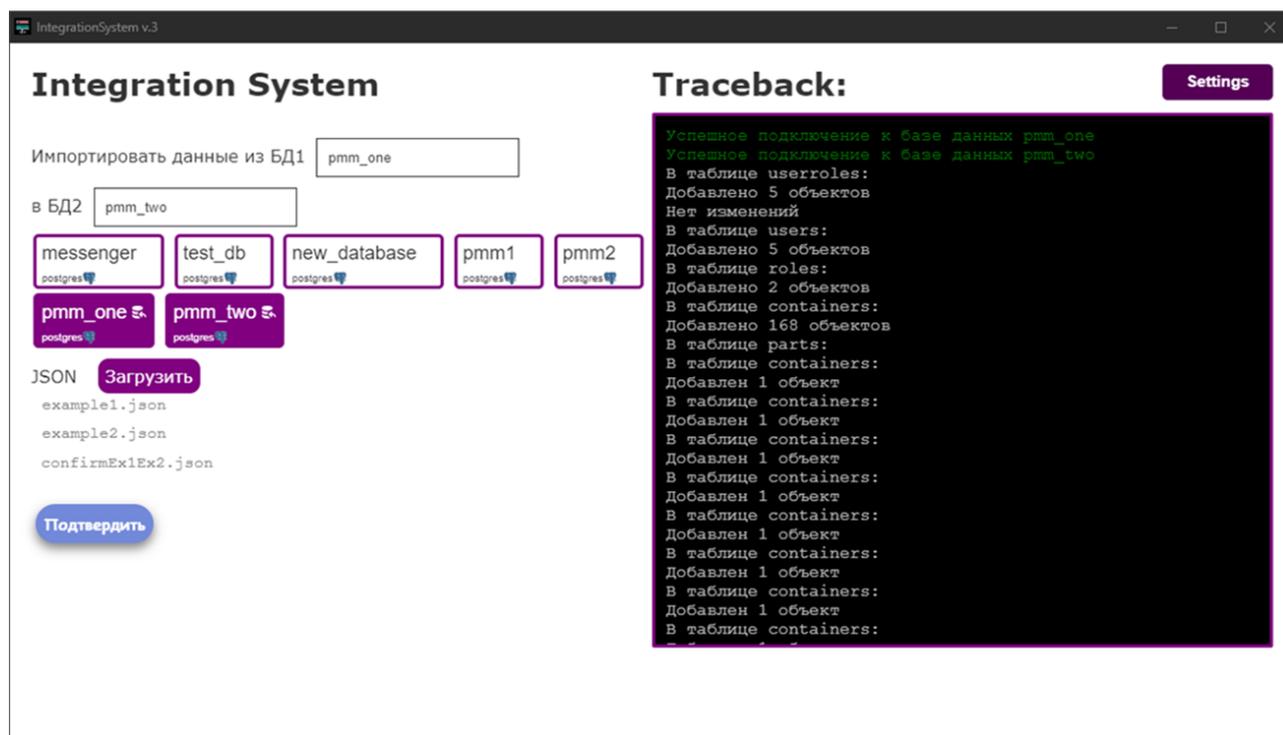


Рисунок 2 – Отчёт об успешном выполнении программы

Интерфейс пользователя, показанный на рисунке 2, предоставляет возможности для настройки подключения к базе данных, загрузки и обработки JSON-файлов, а также выполнения операций с данными в базах. Для начала работы с программой необходимо авторизоваться, указав данные для подключения к PostgreSQL. После успешной авторизации пользователь может настроить имена двух баз данных, с которыми будет работать приложение. На

следующем этапе пользователь может загрузить необходимые JSON-файлы через специальный интерфейс на странице `launch_json`, после чего файлы будут доступны для дальнейшей обработки в программе. После запуска соответствующей операции программа иницирует процессы, необходимые для работы с выбранными данными, результат которых будет выведен в поле `Traceback`. Также интерфейс программы по нажатию кнопки `Settings` предлагает изменить тему оформления или изменить настройки подключения к PostgreSQL.

### 3 Использование JSON в качестве конфигурационных файлов

JSON служит в системе для описания шаблонов, используемых в системе для работы с данными БД. Графический интерфейс сервиса позволяет отредактировать необходимый шаблон (рисунок 3), который затем передаётся на вход в основную программу. Таким образом, JSON является удобным форматом для передачи данных и хорошо поддерживается в различных модулях программы.

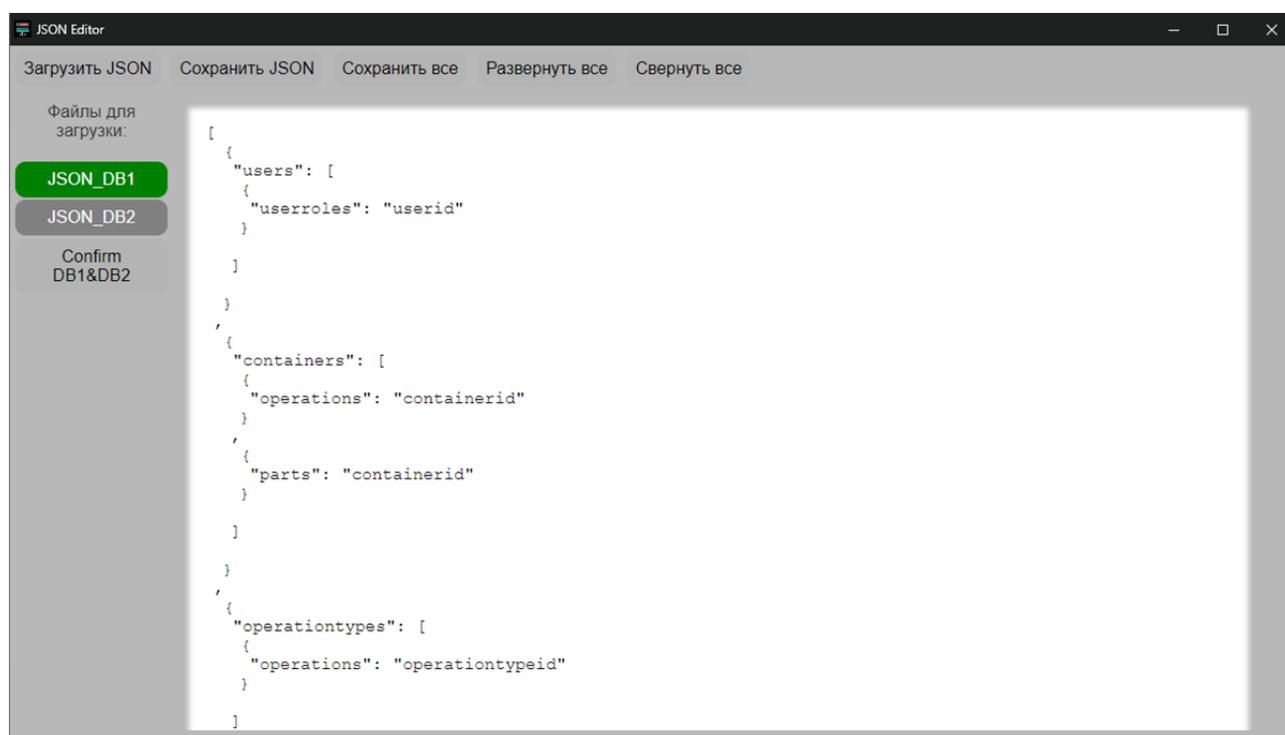


Рисунок 3 – JSON-редактор программы

Используются три конфигурационных файла: перечень таблиц и их полей двух баз данных; перечень соответствия полей двух баз данных. Используя первые два файла, производится загрузка данных из БД, третий файл используется в цикле сравнения загруженного содержимого.

### 4 Тестирование сервиса

Используемая в ходе работы систем информация хранится в реляционных базах данных. В качестве тестового примера база данных была взята из разработанной программной системы, автоматически идентифицирующая детали в процессе производства [4]. Данная си-

стема позволяет автоматизировать процесс получения информации о времени обработки партии деталей, уровне брака, маршруте движения деталей. Для автоматизации используются QR-коды в сочетаниях с обученными нейронными сетями YOLO [5, 6] и VGG19 [7]. Информация в систему поступает с web- или ip-камер. На рисунке 4 приведена ER-диаграмма (сущность-связь) спроектированной для системы базы данных.

Модель данных, содержит 15 сущностей – таблиц базы данных системы. Структура базы данных имеет три группы сущностей: 1) отвечающих за авторизацию и права пользователей; 2) отвечающих за хранение информации по деталям, тарам и операциям над ними; 3) отвечающих за хранение справочной информации по типам деталей, тар, операций, оборудования и структуры предприятия. Например, таблица *Parts* – содержит информацию о деталях; *Containers* – содержит информацию о тарах (партиях однотипных деталей); *Operations* – содержит информацию о технологических операциях над деталями.

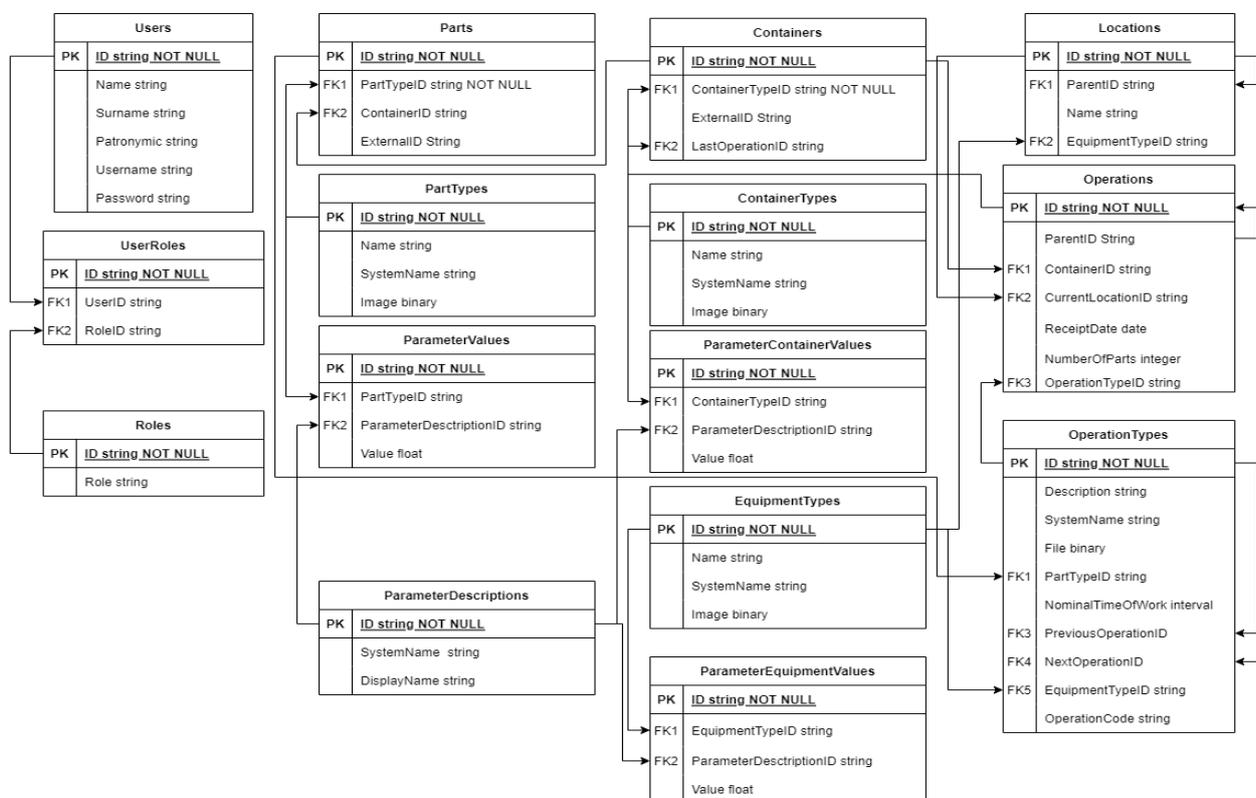


Рисунок 4 – Модель данных для тестирования

Каждый экземпляр каждой сущности имеет идентификатор ID, представляющий собой уникальную строку символов, задаваемую с использованием текстового представления стандарта UUID. Для создания двух баз данных использовался код на Python с использованием SQLAlchemy, позволяющий создать структуру и тестовое наполнение.

Для тестирования были созданы две БД на PostgreSQL: одна для хранения некоторых собранных параметров, вторая – для хранения данных перед отправкой в систему для их дальнейшей обработки. Это позволяло проверить функциональность и надёжность сервиса, что важно для обеспечения работы программы в различных сценариях использования.

В таблице 1 приведены количественные характеристики двух сравниваемых баз данных и время работы сервиса. Время изменяется в зависимости от объёма информации ячеек таблиц

и их количества, а также от особенностей оборудования, поэтому время операций для одних и тех же баз данных может изменяться.

Таблица 1 – Результаты тестирования сервиса

Название таблицы	Удалено (объектов)	Добавлено (объектов)	Обновлено (объектов)	Время выполнения, с
userroles	84	5	0	0,21387
users	0	5	0	0,00399
roles	0	2	0	0,00599
containers	0	168	0	0,00599
parts	0	1080	0	1,64405
operations	0	932	0	4,61232
locations	0	7	0	0,00299
containertypes	0	2	0	0,00699
parttypes	0	4	0	0,00799
equipmenttypes	0	2	0	0,00999
operationtypes	0	22	0	0,03298
parametercontainervalues	0	8	0	0,00600
parametervalues	0	12	0	0,00399
parameterequipmentvalues	0	3	0	0,00399
parameterdescriptions	0	23	0	0,00299
Общее время проведения операций				6,81006

С учётом того, что сервис реализован на языке Python, время выполнения операций по всей базе данных не велико. Для повышения скорости работы может быть применена многопоточность.

### Заключение

В ходе работы был реализован сервис-конвертер данных для систем сбора и передачи информации. Использование подобных программ в составе систем автоматизации предприятий позволит повысить эффективность их использования. Приведённый в статье пример программы отслеживания материальных потоков по цепочке технологического передела, база данных которой использовалась для тестирования разработанного сервиса, является одним из примеров применения используемых разработок. В реалиях Российских предприятий, использующих зачастую разнородные системы («ЛОЦМАН», «АДЕМ», «Диспетчер» и т.п.), использование подобных сервисов для конвертации данных особенно актуально. Дальнейшими направлениями работы в данной области является доработка программного интерфейса, обеспечение совместимости с различными типами баз данных и проведение тестирования сервиса для используемых на производствах системах. Работа выполнена по проекту FSSS-2024-0019, реализуемого в рамках национального проекта «Наука и университеты».

### Список использованных источников

1. О развитии станкоинструментальной промышленности : тезисы доклада заместителя Министра промышленности и энергетики Российской Федерации А. Г. Реуса на заседании Правительственной комиссии по развитию промышленности, транспорта и технологий [Электронный ресурс] // Росбалт [сайт]. – 2007. – URL: <https://www.rosbalt.ru/news/2007-08-02/o-razvitii-stankoinstrumentalnoy-promyshlennosti-3476491> (дата обращения: 02.03.2024).

2. Зотова, Е. А. Интеграция ERP и MES-систем / Е. А. Зотова // Молодой ученый. – 2019. – № 4 (242). – С. 1-3.
3. Андреев, Е. Б. MES-системы: взгляд изнутри / Е. Б. Андреев, И. В. Куцевич, Н. А. Куцевич. – Москва : РТСофт, 2015. – 240 с.
4. Гинятулин, Р. Р. Концепция программной системы для идентификации объектов производственной среды / Р. Р. Гинятулин, Е. Ю. Печенина, В. А. Печенин // Перспективы развития двигателестроения : материалы международной научно-технической конференции имени Н. Д. Кузнецова. – Самара : Самарский национальный исследовательский университет имени академика С. П. Королева, 2023. – С. 259-260.
5. Redmon, J. You only look once: Unified, real-time object detection / J. Redmon, S. Divvala, R. Girshick, F. Farhadi // 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). – 2016. – P. 779-788.
6. Redmon, J. YOLOv3: An Incremental Improvement / J. Redmon, A. Farhadi // arXiv. – 2018. – Vol. 1804. – 02767.
7. Simonyan, K. Very Deep Convolutional Networks for Large-Scale Image Recognition / K. Simonyan, A. Zisserman. // 3rd International Conference on Learning Representations. – 2015. – Vol. 1409. – P. 1556.

## Development of a data exchange service between software systems

<b>K. D. Tandalov</b>	Student, Research Assistant at Scientific-Research Laboratory «Artificial Intelligence in Production Systems»; Samara National Research University, Samara, Russian Federation; tkd186@mail.ru
<b>V. A. Pechenin</b>	Candidate of Science (Engineering), Associate Professor, Head of Scientific-Research Laboratory «Artificial Intelligence in Production Systems»; Samara National Research University, Samara, Russian Federation; pechenin.va@ssau.ru

*This article presents the development of a service information model for data exchange between ERP and MES systems used for planning and control of production processes. The purpose of the service is the efficient transfer and structuring of information between databases in the production process, as well as control of operations through the user interface. This approach allows for the integration of various external software systems, which is an urgent task in the field of automation and control of production processes. The service was implemented in Python using SQLAlchemy to work with PostgreSQL databases. The graphical interface of the service allows you to edit the required JSON template, used for working with databases. Testing was carried out on two databases for a system for monitoring the movement of parts during the production process, confirming the functionality and reliability of the service.*

**Keywords:** information model; system; JSON; databases; Python

**Citation:** Tandalov, K. D. and Pechenin, V. A. (2024), "Development of a data exchange service between software systems", *Journal of Dynamics and Vibroacoustics*, vol. 10, no. 3, pp. 50-57. DOI: 10.18287/2409-4579-2024-10-3-50-57. (In Russian; abstract in English).

---

---

### References

1. The official site of Rosbalt (2007), "O razvitii stankoinstrumental'noy promyshlennosti : tezisy doklada zamestitya Ministra pro-myshlennosti i energetiki Rossiyskoy Federatsii A. G. Reusa na zasedanii Pravitel'stvennoy komissii po razvitiyu promyshlennosti, transporta i tekhnologiy" [On the development of the machine tool industry : abstracts of the report by Deputy Minister of Industry and Energy of the Russian Federation A. G. Reus at the meeting of the Government Commission for the Development of Industry, Transport and Technologies] [Electronic resource], available at: URL: <https://www.rosbalt.ru/news/2007-08-02/o-razvitii-stankoinstrumentalnoy-promyshlennosti-3476491> (access date: 03/02/2024). (In Russian).
2. Zotova, E. A. (2019), "Integration of ERP and MES systems", *Young scientist*, no. 4 (242), pp. 1-3. (In Russian).
3. Andreev, E. B., Kutsevich, I. V. and Kutsevich, N. A. (2015), *MES-sistemy: vzglyad iznutri* [MES systems: a view from the inside], RTSoft, Moscow, 240 p. (In Russian).
4. Ginnyatulin, R. R., Pechenina, E. Yu. and Pechenin, V. A. (2023), "The concept of software system for identification of objects in the production environment", *Prospects for the development of engine building: materials of the international scientific and technical conference named after N. D. Kuznetsova*, Samara National Research University, Samara, pp. 259-260. (In Russian).
5. Redmon, J., Divvala, S., Girshick, R. and Farhadi, F. (2016), "You only look once: Unified, real-time object detection", *2016 IEEE Conf on Computer Vision and Pattern Recognition (CVPR)*, pp. 779-788.
6. Redmon, J. and Farhadi, A. (2018), "YOLOv3: An Incremental Improvement", *arXiv*, vol. 1804.02767.
7. Simonyan, K. and Zisserman, A. (2015), "Very Deep Convolutional Networks for Large-Scale Image Recognition", *3rd International Conference on Learning Representations*, vol. 1409, p. 1556.